

SDEV1201

LESSON 21

NORMALIZATION

Let's review

Normalization Review

Step 0: Create the initial table

1. Identify Primary Key and repeating group(s) of attributes.

Step 1: 1st Normal Form (1NF) - No more than 2 entities/tables

2. A table must contain only **atomic** attributes.
3. A table **cannot contain any repeating groups of attributes**.

Step 2: 2nd Normal Form (2NF)

4. All non-key attributes in a table must **fully depend on the primary key** of the table. – A violation can only occur in a table with a composite key. This is **only applicable to tables with a composite primary key**.

Step 3: 3rd Normal Form (3NF)

1. A non-key attribute **cannot be dependent** on another non-key attribute.

Identifying Primary Keys, Repeating Groups, Partial and Transitive Dependencies

Choosing PK's

- When choosing a PK it must uniquely identify at least one other attribute in the table.
- With the exception of the Initial Table, the PK must uniquely identify ALL the other attributes in the table.
- Almost every PK attribute will contain the words ID, Number or Code in their name.

Identifying Primary Keys, Repeating Groups, Partial and Transitive Dependencies

Repeating Groups

- For a given value of the PK, is there more than one value for an attribute?
 - For example, for a given StudentID(PK), is there more than one FirstName? A student only has one first name, so FirstName is not a repeating group.
 - However, for a given StudentID, is there more than one CourseID? A student can take more than one CourseID. Therefore, CourseID is a repeating group!
- Imagine this scenario. Someone gives you the value for the PK and asks you for the value of one of the attributes. If you cannot answer them with only one value for that attribute, then it is a repeating group!
- Attributes in a table must be able to be uniquely identified (only one value) by the PK. If they cannot, there are a repeating group and must be removed (1NF)

Identifying Primary Keys, Repeating Groups, Partial and Transitive Dependencies

Partial Dependencies

- Remember that if there is no composite PK, there are no Partial Dependencies! Work smarter, not harder!
- Does the attribute need values for the entire PK to identify a single value for an attribute? If a single value for the attribute can be looked up by providing a value for only part of the PK then it is a Partial Dependency.
 - For example, say you have a composite PK made up of StudentID and CourseID. You also have CourseName, CourseCost and FinalGrade as attributes. Ask yourself:
 - Can a single CourseName, CourseCost, or FinalGrade be looked up by just StudentID OR CourseID? If they can then that attribute is a Partial Dependency!
- Attributes in a table must need values for the entire PK to uniquely identify them. If they do not, there are partial dependencies and must be removed. (2NF)

Identifying Primary Keys, Repeating Groups, Partial and Transitive Dependencies

Transitive Dependencies

- Remember that if there are not at least 2 non-PK attributes(we will call them regular attributes), there are no Transitive Dependencies! Work smarter, not harder is still the best approach!
- Are there any regular attributes that can uniquely identify other regular attributes in the table? If so, then there is a transitive dependency.
 - Remember, since PK names almost always contain ID, Number or Code, look for those keywords!
- For example, say you have a table with OrderNumber(PK), CustomerNumber, FirstName, LastName, OrderDate, Total.
- Since CustomerNumber can uniquely identify one FirstName and one LastName, there is a Transitive Dependency.
- Attributes in a table must be uniquely identified by the PK ONLY. If regular attributes can uniquely identify an attribute there is a transitive dependency and must be removed. (3NF)

Identifying Primary Keys, Repeating Groups, Partial and Transitive Dependencies

A couple things to remember

- Foreign keys are treated like a regular attributes in the normalization process. This means that a FK could be a repeating group, part of Partial Dependency or part of a Transitive Dependency.

Once an attribute has been identified as a PK or a FK, it never loses that designation!

The normalization process

1 Collect data

1. Collect **views** (subsets of data) using source documents and business rules.

- Before applying the rules, you should have a clear idea of the entities, relationships, and candidate keys.

2 Normalize

2. Using a view, apply the rules of 1NF, 2NF, and 3NF to produce a **schema**.

- Repeat step #2 for each view.

3 Merge

3. **Merge** all those schemas into a single schema.

- Tables with the **same** PK can be merged together. All entities, attributes, and relationships need to be preserved.

Today's Objective

Emergency Services & Product Exercise Suite

- ESP1, ESP2, ESP 3 and ESP 4.
 - Produce 3NF ERDs
 - Merge views

On the Brightspace site see “ESP 2 Specifications”, “ESP 3 Specifications” (Lesson 5) and “ESP 4 Specifications” in “Week 13- Normalization” under Activities.

Note: for now, for creating diagrams please use: <https://draw.io>

Assignment 4 will be available on Brightspace tomorrow on Thursday, November 20, 2025. It is due Friday December 5, 2025 at 5:00 PM.

Important: Quiz 4 on Stored Procedures and Triggers will be next **Monday, November 24, 2025.**

ESP 1 Review

Database Management: Normalization Exercise Suite

The **Emergency Service & Product (ESP)** Company is a very small parts and service company. It currently uses a crude paper based system (3 by 5 cards, paper journal, etc) to keep track of its customers, inventory, suppliers, payments and invoices.

The company is growing rapidly and has found that it is unable to keep pace with the volume of data it must record about its business activities. It has decided that it wishes to automate some portions of its business.

You have been hired to assist the company in analyzing its business activities and to design a set of relational tables that they can use to develop a database system.

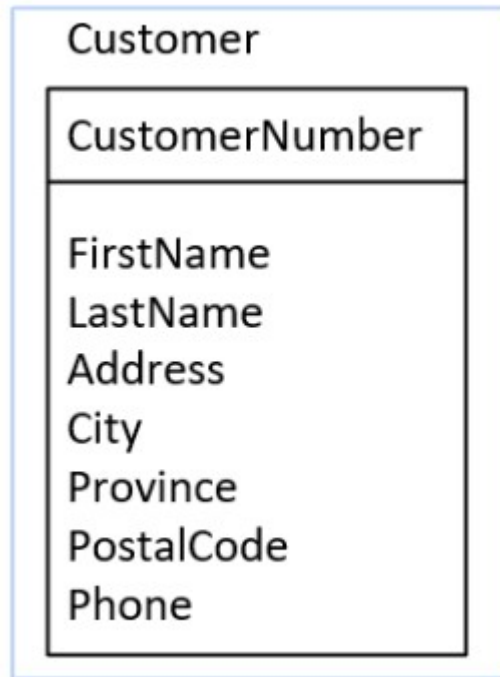
The company will provide you with copies of all their paper based forms (invoices, orders, batch records, etc). You will use these over several weeks to analyze the various forms and produce **third normal form (3NF)** relational **entity relationship diagrams (ERD)** for each **view** of their data. As a last step in the process, you will **merge** all the views to produce one integrated ERD and **fully detailed relational entities**.

Customer Details View

ESP keeps detailed information about each customer so that they can be contacted when needed or when a product needs to be sent in the mail. This information is kept on 3x5 cards as it is fairly static in size and doesn't change very often. The cards look as follows:

Customer Details	Customer Number: 137
Name:	Fred Smith
Address:	123 SomeWhere St.
City:	Edmonton
Province:	Alberta
Postal Code:	T5H 2J9
Home Phone:	436 - 7867

Customer Details View



3NF

CustomerNumber, **FirstName**, **LastName**,
Address, City, Province, PostalCode,
HomePhone

Customer Orders View



Customer Fred Smith

Customer Number: 137

Name:

Address: 123 SomeWhere St.
Edmonton, Ab
T5H 2J9

Phone: 436 - 7867

Date: Jan 16, 2000

Order Number: 219

Item Number	Description	Quantity	Price	Amount
H23	Heater Fan Belt – 23"	1	11.99	11.99
H319	Heater Fan Belt Support Brackets	2	4.99	9.98
M24	Bolts – 24 mm	8	0.29	2.32
			Subtotal:	24.29
			GST:	1.70
			Total:	25.99

ESP has preprinted order forms that they fill in by hand when a customer buys some of their product. The customer details are copied from the customer card onto the order form as well as details about the product ordered. Each order form has a preprinted sequential order number that is unique. The top portion of the forms looks as follows.

Because prices change on a fairly regular basis ESP would like to keep the following information:

- The current price for each item in inventory
- The price each item sold for at the

Customer Orders View

3NF

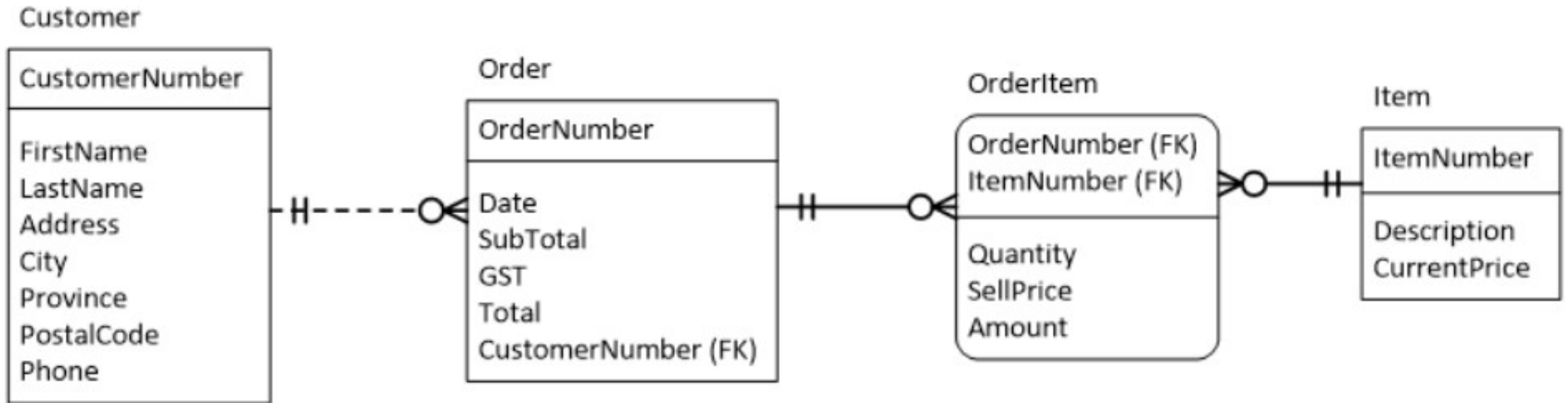
CustomerNumber (FK), Date, **OrderNumber (PK)**, Subtotal, GST, Total

CustomerNumber (PK), CustomerFirstName, LastName, Address, City, Province, PostalCode, Phone

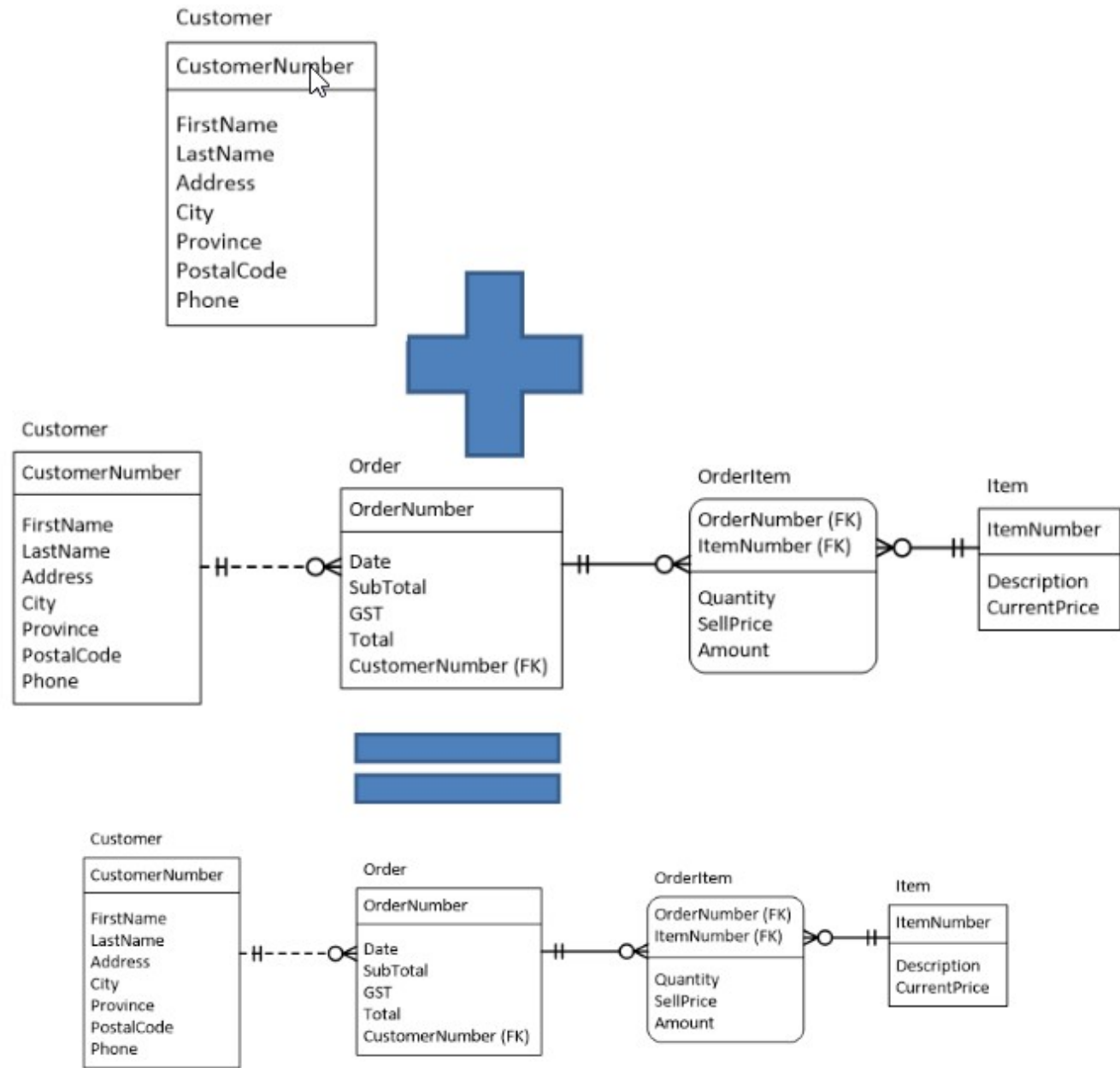
OrderNumber(PK)(FK), **ItemNumber (PK)(FK)**, Quantity, SellingPrice, Amount

ItemNumber (PK), Description, CurrentPrice

Customer Orders View



Merge



ESP 2

Available in Brightspace under “Week
13 – Normalization” in Activites

Customer Payments Against Orders

Customer Payments Against Orders

ESP allows its customers to make payments against their orders in several ways. These include:

- Full payment at the time of order
- Partial payment at the time of order with final payment(s) at a later time
- No payment at time of order with final payment(s) at a later time
- Payments can be made by cash, cheque, Visa, America Express, etc

ESP keeps a payment log to keep track of all payments made against an order. Each order has its own log sheet. The log looks similar to the following:

Payments Log			Order Number: <u>123</u>		
Customer Name: <u>David Williams</u>			Order Date: <u>Jan 11/2019</u>		
Customer Number: <u>78</u>			Order Total: <u>\$145.18</u>		
Date	Payment Amount	Payment Number	<u>Balance Owing</u>	Payment Type	Deposit Batch #
Jan 11/99	100.00	1	45.18	Cheque	118
Jan 18/99	45.18	2	0.00	Cash	119

Entities

Order

Customer

Order Payment

Customer Payments Against Orders *

Note that a customer payment is applied against a single order.

When a payment is made, ESP goes through the following process (worst case scenario – where a cheque arrives in the mail with no indication of an order number on it):

- Customer name is searched in the customer file to determine their customer number. The address on the cheque is used to confirm the customer.
- The Orders File and the Payments Log are manually searched by customer number to find any order(s) that have an outstanding balance.
- If this is the first payment for this order, then a new Payment Log form is filled in with customer name, number, etc.
- The payment details are entered and the balance owing is updated.
- The Deposit Batch # is entered on the log when the payment is deposited into the bank so that the payments can be reconciled with the deposit on the bank statement.

ESP has indicated that there are never any over payments made and refunds or cancellations are never issued.

ESP has also indicated that they would like the new system to only allow specific payment types so they can monitor usage of each type.

You are to take the information provided above and prepare a set of 3NF relational entities and produce a relational 3NF ERD.

Customer Payments Against Orders *

Initial Table

OrderNumber (PK), OrderDate, OrderTotal, CustomerName, CustomerNumber, (PaymentDate, PaymentAmount, PaymentNumber, BalanceOwing, PaymentType, DepositBatchNumber)

Customer Payments Against Orders *

1NF

OrderNumber (PK), OrderDate, OrderTotal,
CustomerFirstName, CustomerLastName, CustomerNumber

OrderNumber (PK)(FK), **PaymentNumber (PK)**,
PaymentDate, PaymentAmount, BalanceOwing, PaymentType,
DepositBatchNumber

Customer Payments Against Orders *

2NF - No Change

OrderNumber (PK), OrderDate, OrderTotal , CustomerFirstName, CustomerLastName, CustomerNumber

OrderNumber (PK)(FK), PaymentDate, PaymentAmount, **PaymentNumber(PK)**, BalanceOwing, PaymentType, DepositBatchNumber

NOTE: An order can have multiple payments. Each order will have a payment number of at least 1 and many will have payments of 1, 2, 3 and all with different payment dates, amounts, types, balance owings and deposit branch numbers. All these attributes are thus dependent on both the order number and the payment number.

Customer Payments Against Orders

3NF

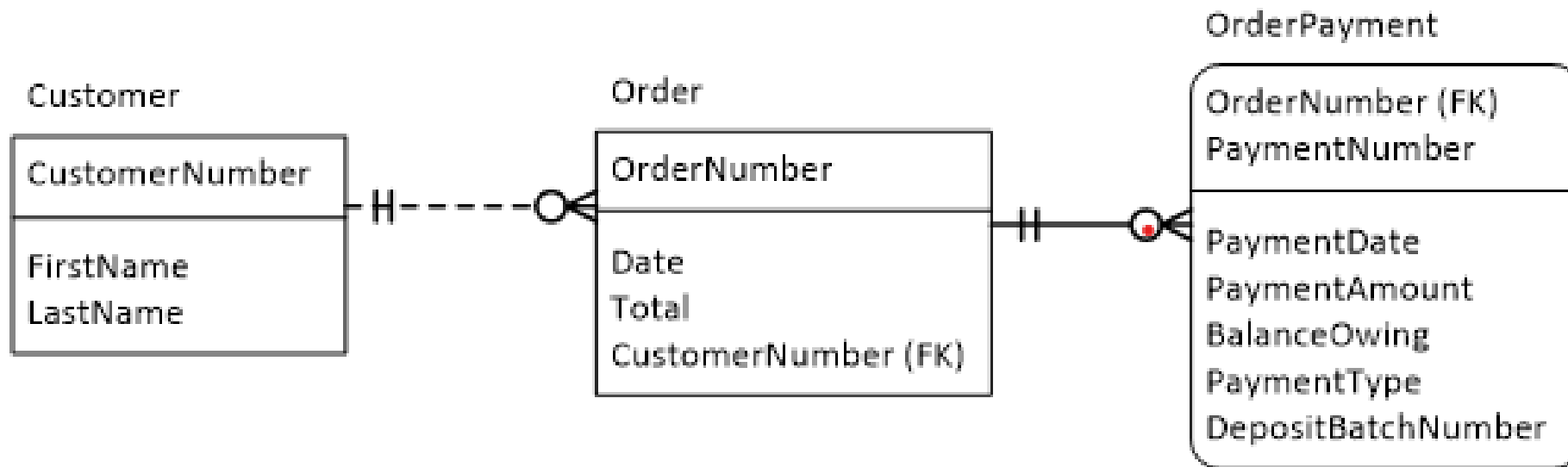
OrderNumber (PK), OrderDate, OrderTotal, *CustomerNumber (FK)*

CustomerNumber (PK) , CustomerFirstName, CustomerLastName

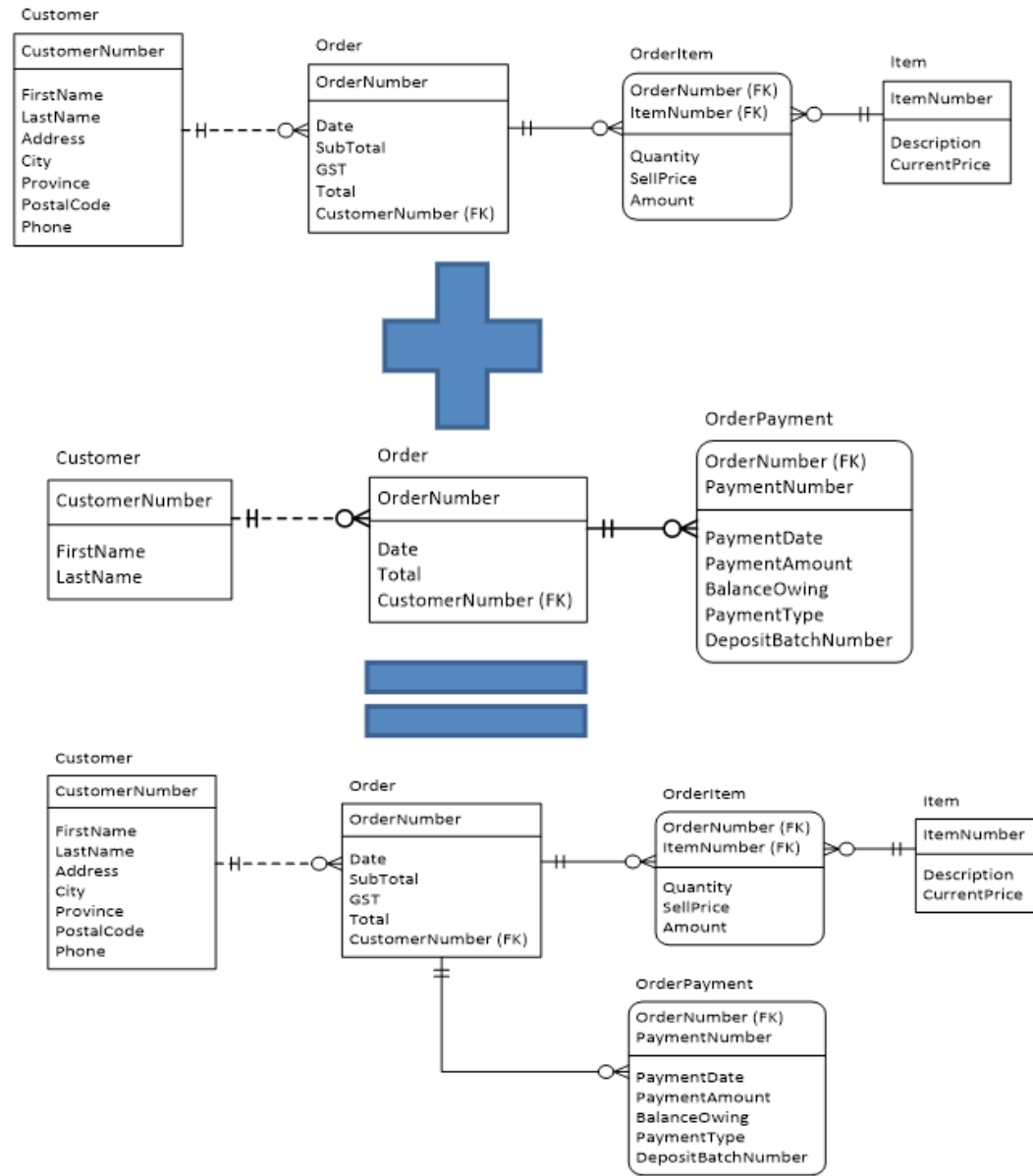
OrderNumber (PK)(FK), PaymentDate, PaymentAmount, **PaymentNumber (PK)**, BalanceOwing, PaymentType, DepositBatchNumber

Note: In the OrderPayment table, BalanceOwing is not dependent on PaymentAmount. It is a calculated attribute which the OrderTotal needs to be known. OrderTotal resides in the Order table. BalanceOwing is dependent on both OrderNumber and PaymentNumber.

Customer Payments Against Orders



Merge



ESP 3

Available in Brightspace under
“Week 13 – Normalization” in
Activites

Inventory Control View *

ESP would like to be able to keep track of how many of each item they have in stock on an on-going basis so that they can use JIT (Just In Time) ordering rather than rather than ordering when they run out. They would like the new system to keep an in-stock count for each item as well as a re-order value. If the in-stock count is less than or equal to the re-order level then the item will be ordered from a supplier

Inventory Control Sheet

Item #: M24

Item Description: Bolts – 24 mm

Current Sale Price: \$ 0.29

Order History

<i>Date</i>	<i>Supplier Number</i>	<i>PO Number</i>	<i>Quantity</i>	<i>Cost</i>
July 1, 1998	7	125	1000	0.13
August 2, 1998	10	131	2000	0.13
September 5, 1998	7	149	2000	0.14

Inventory Control View *

Initial Table

ItemNumber, ItemDescription, CurrentSalePrice, QOH, ReorderLvl (OrderDate, SupplierNumber, PONumber, Quantity, Cost)

Note: QOH and ReorderLvl are new attributes that have been requested in the requirements.

Inventory Control View *

1NF

ItemNumber (PK), ItemDescription, CurrentSalePrice, QOH,
ReorderLvl

PONumber (PK), *ItemNumber (PK)(FK)*, OrderDate,
SupplierNumber, Quantity, Cost

Inventory Control View *

2NF

ItemNumber (PK), ItemDescription, CurrentSalePrice, **QOH**,
ReorderLvl

PONumber (PK)(FK), Quantity, Cost, ***ItemNumber (PK)(FK)***

PONumber, OrderDate, SupplierNumber

Inventory Control View

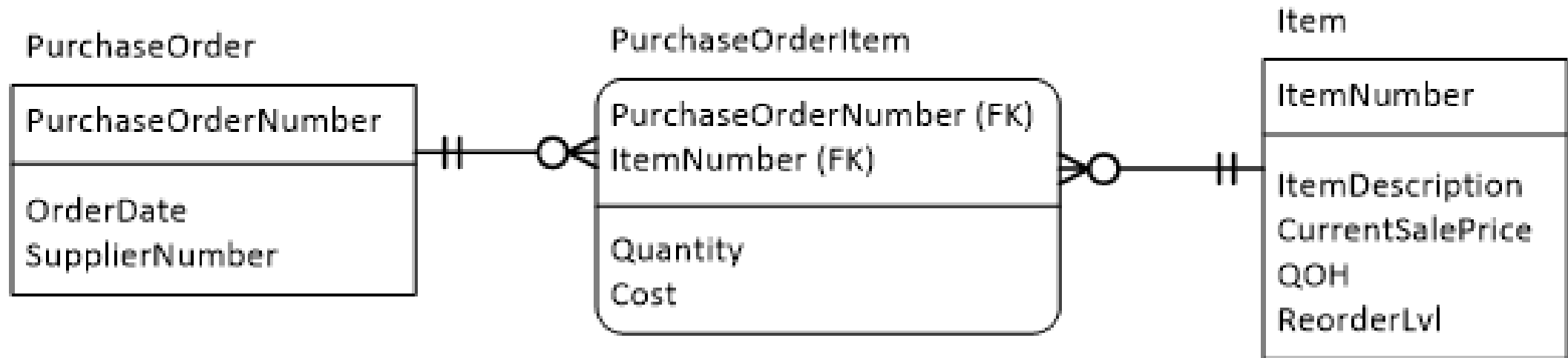
3NF – No change from 2NF

ItemNumber (PK), ItemDescription, CurrentSalePrice, QOH,
ReorderLvl

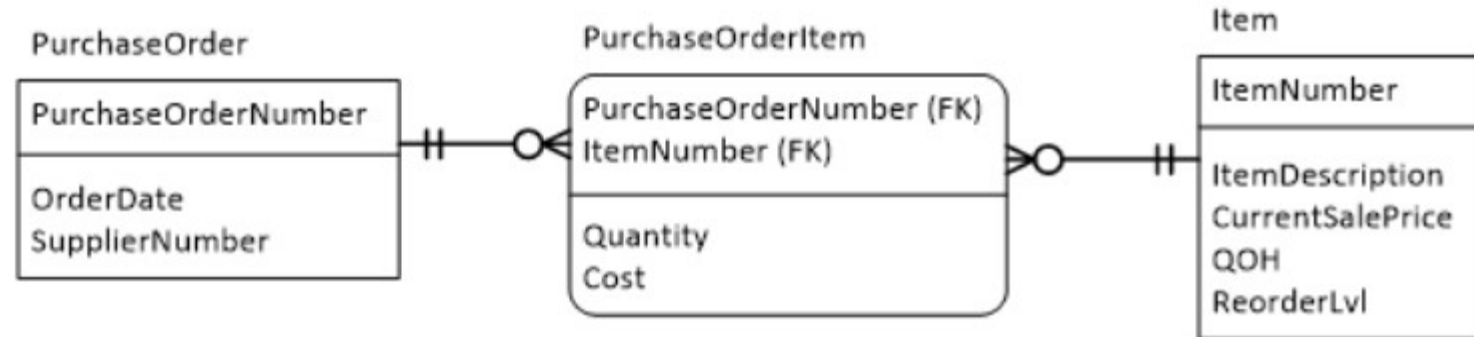
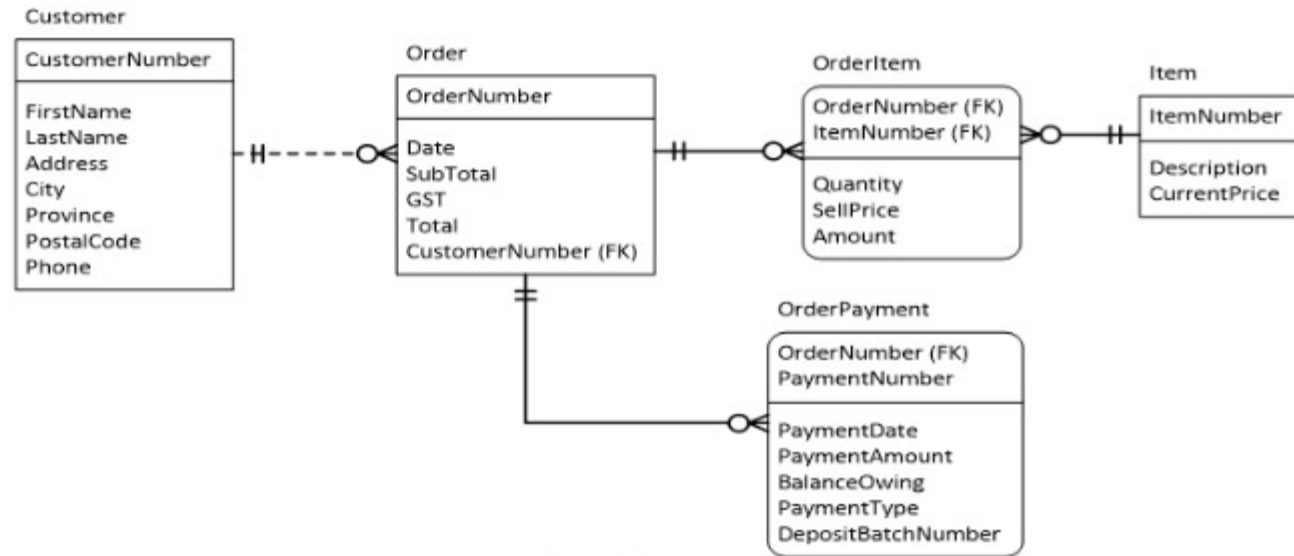
PONumber (PK)(FK), Quantity, Cost, ***ItemNumber (PK)(FK)***

PONumber, OrderDate, SupplierNumber

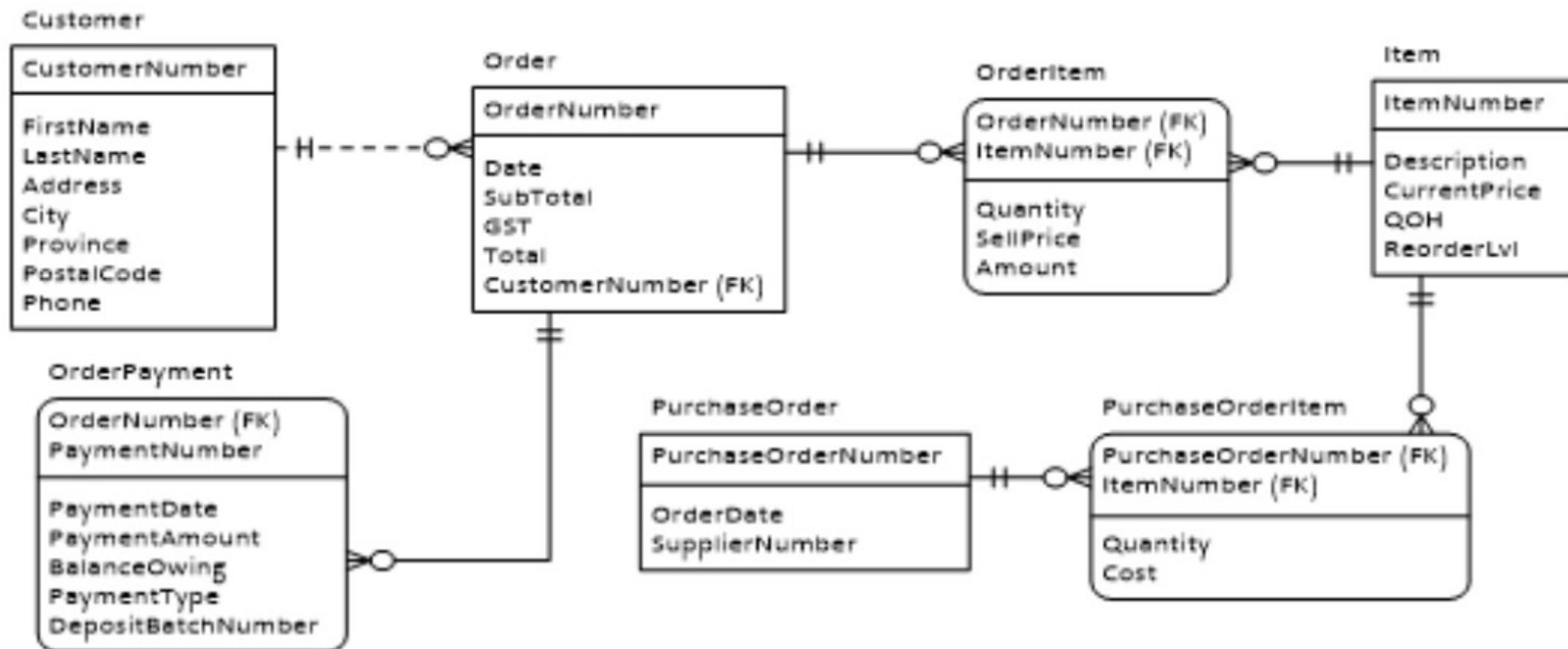
Inventory Control View



Merge



Merge



Purchase Orders *



Purchase Order #125

Supplier Name: Nuts and Bolts Supplier Number: 7
Address: 11123 – 114 Avenue
City: Edmonton, AB, T6G 4X1
Phone: 488-2312 Date: July 1, 1998

Item#	Supplier's Item #	Supplier's Description	Qty	Cost	Amount
H23	S103	Heater Fan Belt – 23"	1	11.99	11.99
H319	B11a	Heater Fan Support Brackets	2	4.99	9.98
M24	ISO4	Bolts – 24 mm	1000	0.13	130.00
				SubTotal:	151.97
				GST:	10.64
				TOTAL:	162.61

Purchase Order (PO) numbers are unique and never reused. The same supplier always has the same supplier number (which is also unique) and the supplier details are hand-written on the PO every time.

The item number is the number assigned by ESP from the Inventory Control Sheet and is the same regardless of supplier the item is purchased from. The Supplier's Item Number and Description are specific to the supplier and may vary on different Purchase Orders. The cost of the item depends on the supplier and may vary between different purchase orders for the same

Purchase Orders *

Initial Table

PurchaseOrderNumber (PK), Date, Subtotal, GST, Total,
SupplierNumber, SupplierName, SupplierAddress, SupplierCity,
Phone, (ItemNumber, SupplierItemNumber,
SuppliersDescription, Qty, Cost, Amount)

Purchase Orders *

1NF

PurchaseOrderNumber (PK), Date, Subtotal, GST, Total,
SupplierNumber, SupplierName, SupplierAddress, *SupplierCity*,
SupplierProvince, *SupplierPostalCode*, Phone

PurchaseOrderNumber (PK)(FK), **ItemNumber (PK)**,
SupplierItemNumber, SuppliersDescription, Qty, Cost, Amount

Purchase Orders *

2NF – No change from 1NF

PurchaseOrderNumber (PK), Date, Subtotal, GST, Total,
SupplierNumber, SupplierName, SupplierAddress, SupplierCity,
SupplierProvince, SupplierPostalCode, Phone

PurchaseOrderNumber (PK)(FK), ItemNumber (PK),
SupplierItemNumber, SuppliersDescription, Qty, Cost, Amount

Purchase Orders

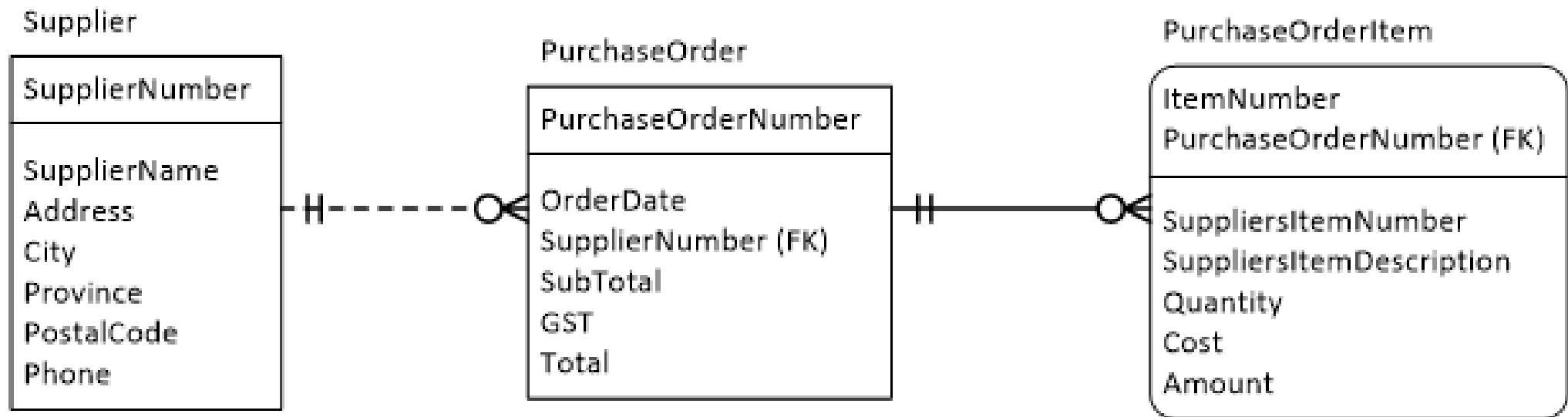
3NF

PurchaseOrderNumber (PK), SupplierNumber (FK), Date,
Subtotal, GST, Total

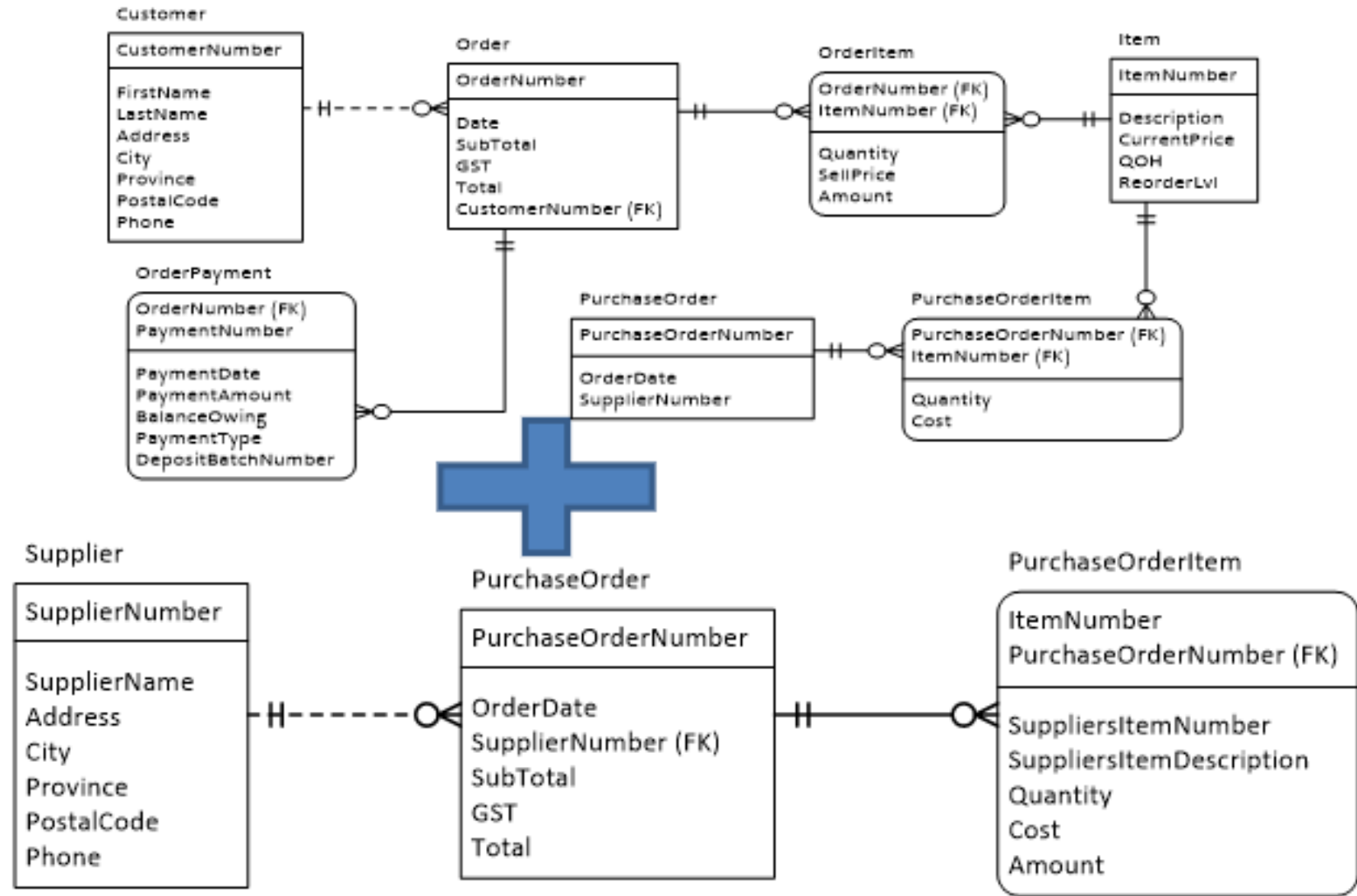
SupplierNumber (PK), SupplierName, SupplierAddress,
SupplierCity,

PurchaseOrderNumber (PK)(FK), **ItemNumber (PK)**,
SupplierItemNumber, SuppliersDescription, Qty, Cost, Amount

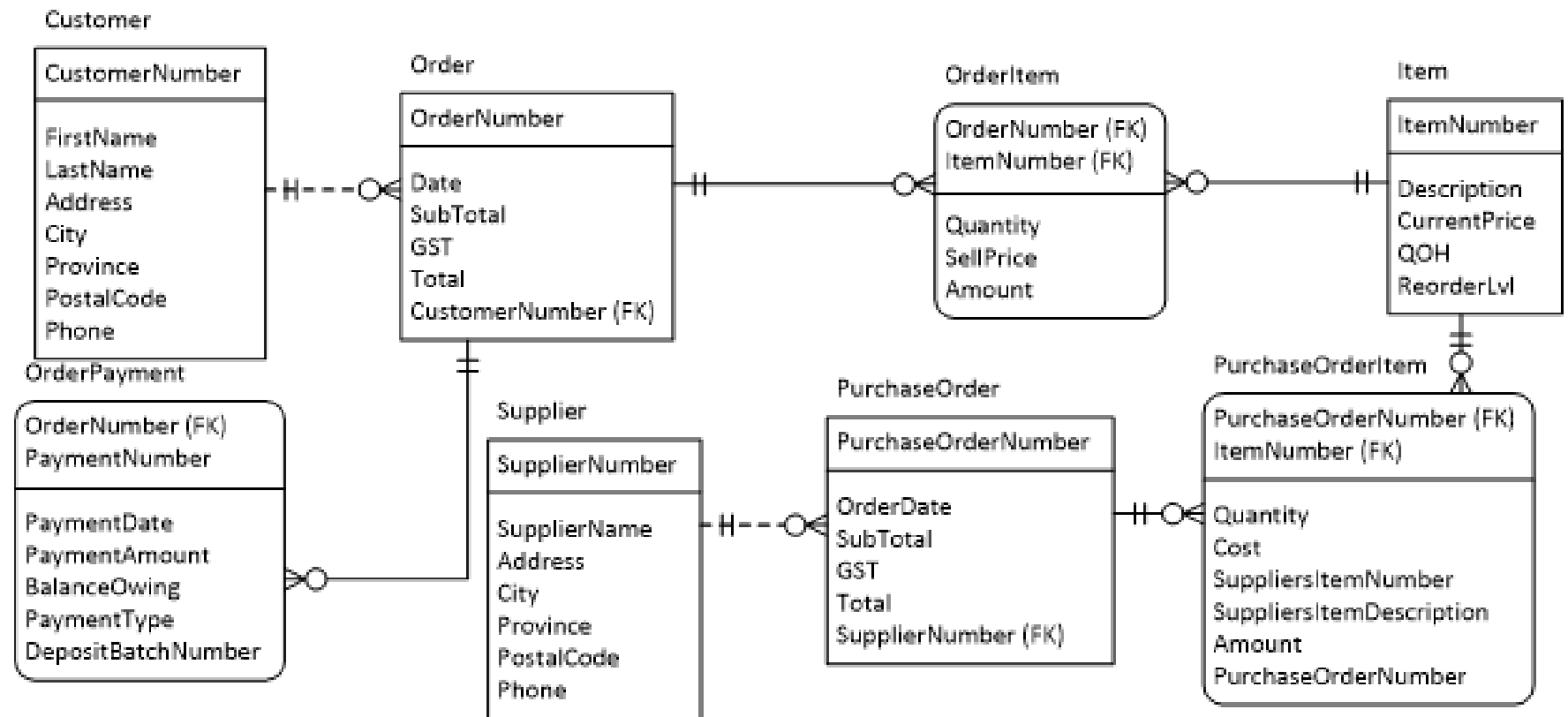
Purchase Orders



Merge



Merge



In the **Purchase Order** View we discovered a new Supplier entity that merge to the existing design through the **Purchase Order** Entity. As well, a number of new attributes for PurchaseOrder and PurchaseOrderItem were discovered.

ESP 4

Available in Brightspace under
“Week 13 – Normalization” in
Activites

Purchase Orders Payments *

ESP pays the full amount of each purchase order when it is due. ESP may pay for more than one purchase order with one cheque if they are all from the same supplier. The cheque is always written for the full and correct amount of all the purchase orders being paid for. ESP keeps a register of its cheques that looks as follows:

<i>Cheque Register</i>						
Date	Cheque #	Cheque Amount	Supplier #	Supplier Name	Purchase Order # (s)	P.O. Date(s)
Jan 11/99	245	\$823.18	7	Nuts & Bolts	79	Dec 23/98
					81	Dec 29/98
Jan 18/99	246	\$427.39	15	Home Hardware	82	Dec 31/98

Each cheque number is unique and numbers are never used over again.

You are to take the information about the Cheque Register, provided above, and prepare a set of **3NF** relational entities and produce a relational **3NF ERD**.

Purchase Orders Payments *

Initial Table

ChequeNumber (PK), ChequeAmount, Date, SupplierNumber,
SupplierName (PurchaseOrderNumber, PurchaseOrderDate)

Purchase Orders Payments *

1NF

ChequeNumber (PK), ChequeAmount, Date, SupplierNumber, SupplierName

PurchaseOrderNumber (PK), *ChequeNumber (FK)*, PurchaseOrderDate

Purchase Orders Payments *

2NF – No change from 1NF

ChequeNumber (PK), ChequeAmount, Date, SupplierNumber, SupplierName

PurchaseOrderNumber (PK), *ChequeNumber (FK)*, PurchaseOrderDate

Purchase Orders Payments

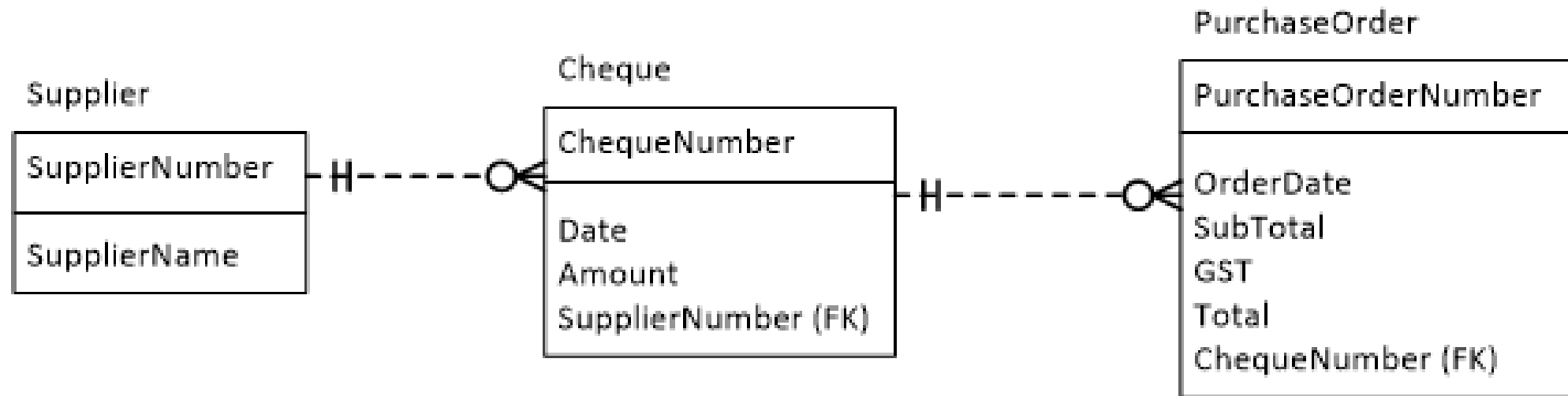
3NF

ChequeNumber (PK), ChequeAmount, Date, *SupplierNumber*
(FK)

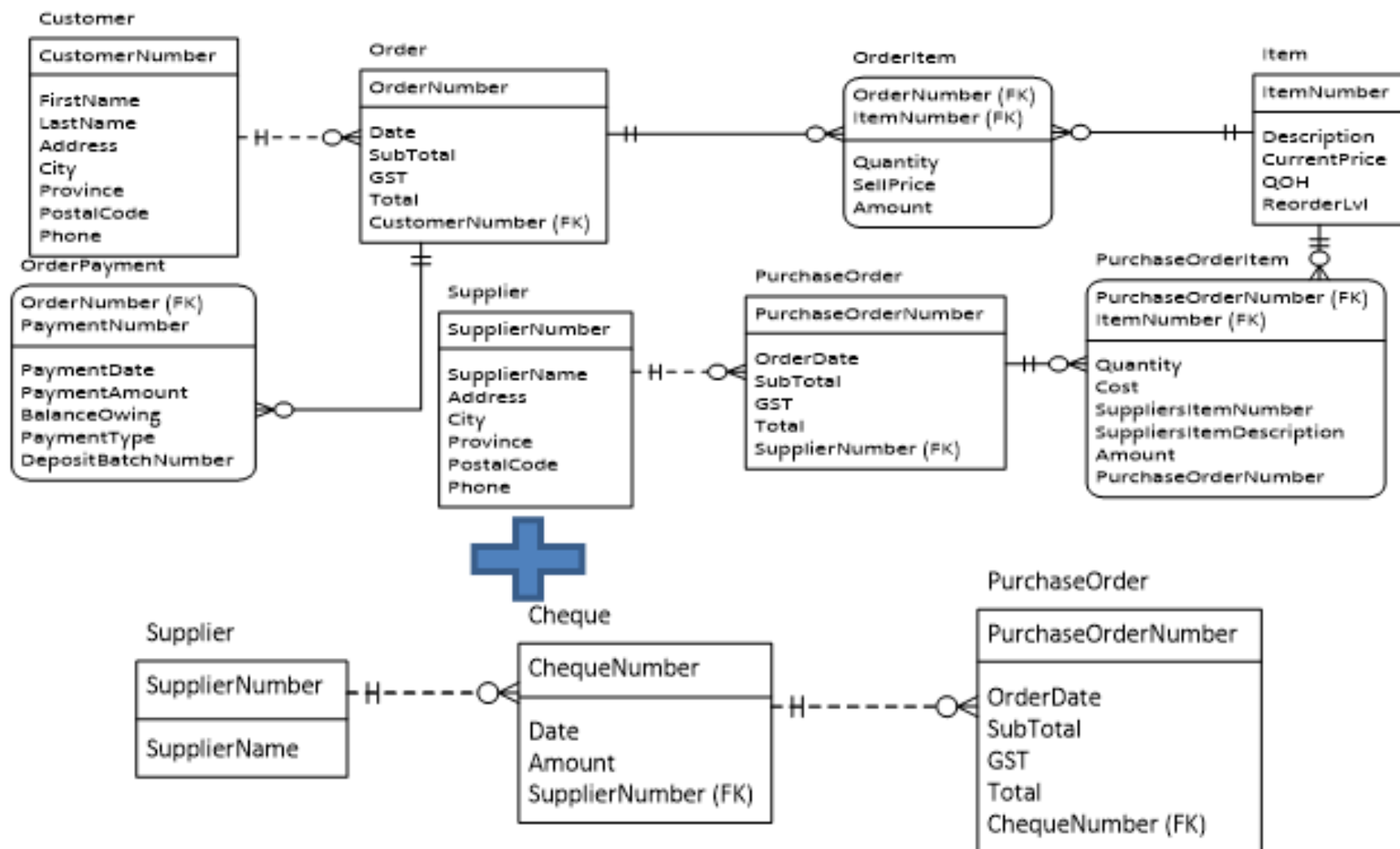
SupplierNumber (PK), SupplierName

PurchaseOrderNumber (PK), *ChequeNumber (FK)*,
PurchaseOrderDate

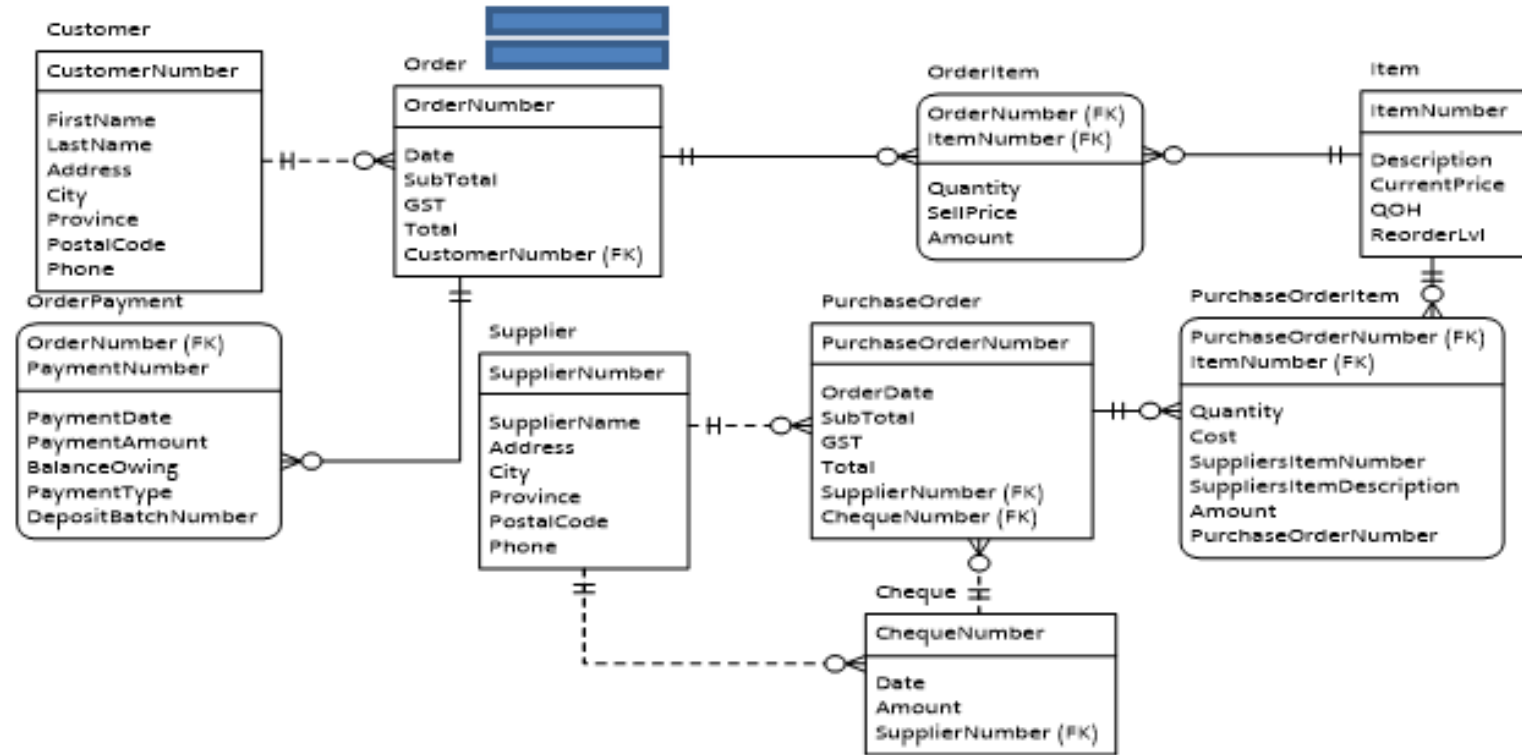
Purchase Orders Payments



Merge



Merge



In the Purchase Order Payments View we discovered a new Cheque entity that merges to the existing design through the Supplier Entity AND the Purchase Order.

This solution introduces a “circle” of relationships between Supplier, PurchaseOrder and Cheque. Depending on the businesses needs one of the relationships (and its corresponding foreign key) might be able to be removed. Could this be possible here?

This is the final database design that will support the operations and business rules as outlined in the six different views.

When doing the normalization procedure

- Follow each step of the normalization process: Initial table, 1NF, 2NF, 3NF. For each steps write out the entity's attributes including the primary key. Only after 3NF can the ERD be created. In Labs and quizzes marks are allotted for all work done for each step and if just the final result is shown marks will be lost for not showing the all of the steps.

This weekend

Also, look at the ESP exercises and take the 3NF tables and practice drawing the ERDs. Do all the views. Then start with the 1st view and merge all other views in numerical order.

Look at <https://draw.io> . It can be used to draw the ERDs which are necessary for the lab.

Here is a brief video on how to use <https://draw.io> :

<https://www.youtube.com/watch?v=JYZPdU5F2iM>

Assignment 4 will be available on Brightspace tomorrow on Thursday, November 20, 2025. It is due Friday December 5, 2025 at 5:00 PM.

Important: Quiz 4 on Stored Procedures and Triggers will be next **Monday, November 24, 2025.**